Power Output Models of Ordinary Differential Equations by Polynomial and Recurrent Neural Networks

Ladislav Zjavka and Václav Snášel

VŠB-Technical University of Ostrava, IT4 Innovations Ostrava, Czech Republic
{ladislav.zjavka,vaclav.snasel}@vsb.cz

Abstract. The production of renewable energy sources is unstable, influenced a weather frame. Photovoltaic power plant output is primarily dependent on the solar illuminance of a locality, which is possible to predict according to meteorological forecasts (Aladin). Wind charger power output is induced mainly by a current wind speed, which depends on several weather standings. Presented time-series neural network models can define incomputable functions of power output or quantities, which direct influence it. Differential polynomial neural network is a new neural network type, which makes use of data relations, not only absolute interval values of variables as artificial neural networks do. Its output is formed by a sum of fractional derivative terms, which substitute a general differential equation, defining a system model. In the case of time-series data application an ordinary differential equation is created with time derivatives. Recurrent neural network proved to form simple solid time-series models, which can replace the ordinary differential equation description.

Keywords: power plant output, solar illuminance, wind charger, differential polynomial neural network, recurrent neural network.

1 Introduction

Power production estimations of renewable sources are necessary as the supplies are very variable [7]. The electrical energy accumulation is an ambitious problem to solve, there is better to consume it direct by customers. A following day output production is a sufficient estimation used by the electrical network operator [8]. The photovoltaic power plant (PVP) or wind charger supply of electricity is difficult to predict using deterministic methods as weather conditions can change from day to day or within short time periods. Hence the power output model should be updated to take into account a dynamic character of applied meteorological variables. Neural networks are able to deal with some problems, which other method solutions fail. They can define simple and reliable models, which exact solution is problematic. Recurrent neural network (RNN) is often used to define models of time-series data applications, which is possible to describe by ordinary differential equations. It applies as inputs also its neuron outputs from a previous time estimate. Analogous to other common neural network solutions, there is not possible to get specifications of RNN models in the form of a math description. The model appears to the users as a “black box”.

A. Abraham et al. (eds.), *Innovations in Bio-inspired Computing and Applications*,
Advances in Intelligent Systems and Computing 237,
DOI: 10.1007/978-3-319-01781-5_1, © Springer International Publishing Switzerland 2014

$$y = a_0 + \sum_{i=1}^{m} a_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij} x_i x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} a_{ijk} x_i x_j x_k + \ldots \qquad (1)$$

m – number of variables $X(x_1, x_2, \ldots, x_m)$ $A(a_1, a_2, \ldots, a_m), \ldots$ *- vectors of parameters*

Differential polynomial neural network (D-PNN) is a new neural network type, which results from the GMDH (Group Method of Data Handling) polynomial neural network (PNN), created by a Ukrainian scientist Aleksey Ivakhnenko in 1968, when the back-propagation technique was not known yet. General connection between input and output variables is possible to express by the Volterra functional series, a discrete analogue of which is Kolmogorov-Gabor polynomial (1). This polynomial can approximate any stationary random sequence of observations and can be computed by either adaptive methods or system of Gaussian normal equations [5]. GMDH decomposes the complexity of a process into many simpler relationships each described by low order polynomials (2) for every pair of the input values. Typical GMDH network maps a vector input $\boldsymbol{x}$ to a scalar output y, which is an estimate of the true function $f(\boldsymbol{x}) = y^t$.

$$y = a_0 + a_1x_i + a_2x_j + a_3x_ix_j + a_4x_i^2 + a_5x_j^2 \qquad (2)$$

D-PNN combines the PNN functionality with some math techniques of differential equation (DE) solutions. Its models are a boundary of neural network and exact computational techniques. D-PNN forms and resolves an unknown general DE description of a searched function approximation. A DE is substituted producing sum of fractional polynomial derivative terms, forming a system model of dependent variables. In contrast with the ANN functionality, each neuron can direct take part in the total network output calculation, which is generated by the sum of active neuron output values. Its function approximation is based on any dependent data relations.

2 Modeling and Forecasting of Energy Production

The potential benefits of having energy production predictability are obvious useful in automatic power dispatch, load scheduling and energy control. The chance to forecast the energy production up to 24 hours can become of the utmost importance in decision-making processes, with particular reference to grid connected photovoltaic plants. Several approaches to forecasting load, wind speed or solar irradiation can be found. They can include neural network regression methods (Auto-Regressive Moving Average Model) and time series analysis models (Nonlinear Autoregressive with Exogenous Input). However most of the existing methodologies show some drawbacks such as high average accuracy error and dependence on the particular design of the PVP. Variability of weather, in particular solar irradiation, is maybe the main difficulty faced by PVP operators so that good forecasting tools are required for the appropriate integration of renewable energy into the power system. Neural networks are able to model the nonlinear nature of dynamic processes, reproduce an empirical, possibly nonlinear, relationship between some inputs and one or more outputs. They are applied for such purpose regarding to its approximation capability of any continuous nonlinear function with arbitrary accuracy that offer an effective alternative to more traditional statistical techniques [3]. Measurements of environmental parameters are generally provided in the form of time series which are suitable to use artificial neural networks with tapped delay lines. Regarding the training window width, the typical way to provide data for solar energy climatology has monthly, annual or 10-days granularity. Several solar parameters might be considered e.g. clearness, visibility index, cloud coverage and sunshine duration however the solar radiation is the most important parameter in the prediction and modeling of renewable PVP energy systems [2].

The wind speed model can apply 2 different inputs: lagged values of the average or maximum wind speed. The wind energy and speed change are not continual throughout the entire year, for this reason might be used the wind velocity meteorological maps providing regional assessments and interpretations. The difficulty in predicting this meteorological parameter arises from the fact that it is a result of the complex interactions among large-scale forcing mechanisms of pressure and temperature differences, local characteristics of the surface, etc. Wind energy is strong especially during winter, in the period with the highest demand. End-users recognize the contribution of wind prediction for a secure and economic operation of the power system. The power models should take into account technical parameters, i.e. hub height, turbine type, etc. Wind energy is possible to forecast using neuro-fuzzy, cognitive mapping or other soft computing techniques [1].

3 General Differential Equation Composition

The basic idea of the D-PNN is to compose and substitute a general sum partial differential equation (3), which is not known in advance and can describe a system of dependent variables, with a sum of fractional relative multi-parametric polynomial derivative terms (4).

$$a + \sum_{i=1}^{n} b_i \frac{\partial u}{\partial x_i} + \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \ldots = 0 \qquad u = \sum_{k=1}^{\infty} u_k \tag{3}$$

$u = f(x_1, x_2, \ldots, x_n)$ – searched function of all input variables
$a, B(b_1, b_2, \ldots, b_n), C(c_{11}, c_{12}, \ldots)$ – polynomial parameters

Partial DE terms are formed according to the adapted integral analogues method, which is a part of similarity model analysis. It replaces mathematical operators and symbols of a DE by ratio of corresponding values. Derivatives are replaced by their integral analogues, i.e. derivative operators are removed and simultaneously with all operators are replaced by similarly or proportion signs in equations to form dimensionless groups of variables [4].

$$u_i = \frac{\left(a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2 + \ldots\right)^{m/n}}{b_0 + b_1x_1 + \ldots} = \frac{\partial^m f(x_1, \ldots, x_n)}{\partial x_1 \partial x_2 \ldots \partial x_m} \tag{4}$$

n – combination degree of a complete polynomial of n-variables
m – combination degree of denominator variables

The fractional polynomials (4) define partial relations of n-input variables. The numerator of a DE term (4) is a polynomial of all n-input variables and partly defines an unknown function u of eq. (3). The denominator is a derivative part, which includes an incomplete polynomial of the competent combination variable(s). The root function of numerator takes the polynomial into competent combination degree to get the dimensionless values [4]. In a case of time-series data application an ordinary differential equation is formed with only time derivatives. The partial DE (3) might become form of (5).

$$a + bf + \sum_{i=1}^{m} c_i \frac{df(t, x_i)}{dt} + \sum_{i=1}^{m} \sum_{j=1}^{m} d_{ij} \frac{d^2 f(t, x_i, x_j)}{dt^2} + \ldots = 0 \tag{5}$$

$f(t, \mathbf{x})$ – function of time t and independent input variables $\mathbf{x}(x_1, x_2, \ldots, x_m)$

Blocks of the D-PNN (Fig.1.) consist of derivative neurons, one for each fractional polynomial derivative combination, so each neuron is considered a summation DE term (4). Each block contains a single output polynomial (2), without derivative part. Neurons do not affect the block output but participate direct in the total network output sum calculation of a DE composition. Each block has *1* and neuron *2* vectors of adjustable parameters ***a***, resp. ***a, b***.

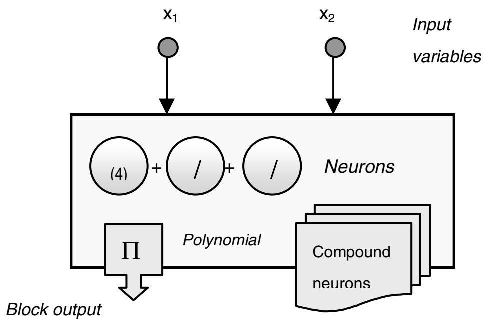


Fig. 1. D-PNN block of basic and compound neurons

In the case of 2 input variables the 2nd odder partial DE can be expressed in the form (6), which involve all derivative terms of variables applied by the GMDH polynomial (2). D-PNN processes these 2-combination square polynomials of blocks and neurons, which form competent DE terms of eq. (5). Each block so include *5* basic neurons of derivatives $x_1, x_2, x_1x_2, x_1^2, x_2^2$ of the 2nd order partial DE (6), which is most often used to model physical or natural systems.

$$F\left(x_1, x_2, u, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, \frac{\partial^2 u}{\partial x_2^2}\right) = 0 \tag{6}$$

where $F(x_1, x_2, u, p, q, r, s, t)$ is a function of 8 variables

4 Differential Polynomial Neural Network

Multi-layered networks forms composite polynomial functions (Fig.2.). Compound terms (CT), i.e. derivatives in respect to variables of previous layers, are calculated according to the composite function partial derivation rules (7)(8). They are formed by products of partial derivatives of external and internal functions.

$$F(x_1, x_2, \dots, x_n) = f(y_1, y_2, \dots, y_m) = f(\phi_1(X), \phi_2(X), \dots, \phi_m(X)) \tag{7}$$

$$\frac{\partial F}{\partial x_k} = \sum_{i=1}^{m} \frac{\partial f(y_1, y_2, \dots, y_m)}{\partial y_i} \cdot \frac{\partial \phi_i(X)}{\partial x_k} \qquad k=1, \dots, n \tag{8}$$

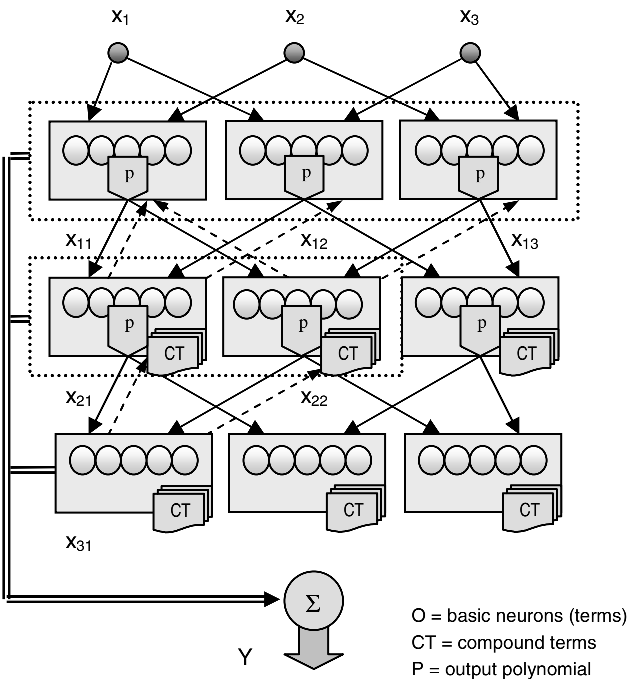


Fig. 2. 3-variable multi-layered D-PNN with 2-variable combination blocks

Thus blocks of the 2nd and following hidden layers are additionally extended with compound terms (neurons), which form composite derivatives utilizing outputs and inputs of back connected previous layer blocks. The 1st block of the last (3rd) hidden layer forms neurons e.g. (9)(10)(11) [9].

$$y_1 = \frac{\partial f(x_{21}, x_{22})}{\partial x_{21}} = w_1 \frac{\left(a_0 + a_1 x_{21} + a_2 x_{22} + a_3 x_{21} x_{22} + a_4 x_{21}^2 + a_5 x_{22}^2\right)^{1/2}}{3/2 \cdot (b_0 + b_1 x_{21})} \tag{9}$$

$$y_2 = \frac{\partial f(x_{21}, x_{22})}{\partial x_{11}} = w_2 \frac{(a_0 + a_1x_{21} + a_2x_{22} + a_3x_{21}x_{22} + a_4x_{21}^2 + a_5x_{22}^2)^{1/2}}{3/2 \cdot x_{22}} \cdot \frac{(x_{21})^{1/2}}{3/2 \cdot (b_0 + b_1x_{11})} \quad (10)$$

$$y_3 = \frac{\partial f(x_{21}, x_{22})}{\partial x_1} = w_3 \frac{(a_0 + a_1x_{21} + a_2x_{22} + a_3x_{21}x_{22} + a_4x_{21}^2 + a_5x_{22}^2)^{1/2}}{3/2 \cdot x_{22}} \cdot \frac{(x_{21})^{1/2}}{3/2 \cdot x_{12}} \cdot \frac{(x_{11})^{1/2}}{3/2 \cdot (b_0 + b_1x_1)} \quad (11)$$

The square (12) and combination (13) derivative terms are also calculated according to the composite function derivation rules.

$$y_4 = \frac{\partial^2 f(x_{21}, x_{22})}{\partial x_{11}^2} = w_4 \frac{(a_0 + a_1x_{21} + a_2x_{22} + a_3x_{21}x_{22} + a_4x_{21}^2 + a_5x_{22}^2)^{1/2}}{1.5 \cdot x_{22}} \cdot \frac{x_{21}}{3.8 \cdot (b_0 + b_1x_{11} + b_2x_{11}^2)} \quad (12)$$

$$y_5 = \frac{\partial^2 f(x_{21}, x_{22})}{\partial x_{11}\partial x_{12}} = w_5 \frac{(a_0 + a_1x_{21} + a_2x_{22} + a_3x_{21}x_{22} + a_4x_{21}^2 + a_5x_{22}^2)^{1/2}}{1.5 \cdot x_{22}} \cdot \frac{x_{21}}{3.3 \cdot (b_0 + b_1x_{11} + b_2x_{12} + b_3x_{11}x_{12})} \quad (13)$$

The best-fit neuron selection is the initial phase of the DE composition, which may apply a proper genetic algorithm (GA). Parameters of polynomials might be adjusted by means of difference evolution algorithm (EA), supplied with sufficient random mutations. The parameter optimization is performed simultaneously with the GA term combination search, which may result in a quantity of local or global error solutions. There would be welcome to apply an adequate gradient descent method too, which parameter updates result from partial derivatives of polynomial DE terms in respect with the single parameters [6]. The number of network hidden layers coincides with a total amount of input variables.

$$Y = \frac{\sum_{i=1}^{k} y_i}{k} \qquad k = \textit{amount of active neurons} \quad (14)$$

Only some of all potential combination DE terms (neurons) may participate in the DE composition, in despite of they have an adjustable term weight (w_i). D-PNN's total output Y is the sum of all active neuron outputs, divided by their amount k (14).

$$E = \sqrt{\frac{\sum_{i=1}^{M} \left(y^d - y_i\right)^2}{M}} \rightarrow \min \quad (15)$$

The root mean square error (RMSE) method (15) was applied for the polynomial parameter optimization and neuron combination selection. D-PNN is trained only with a small set of input-output data samples likewise the GMDH algorithm does [6].

5 Power Plant Output Model Experiments

D-PNN and RNN (Fig.7.) apply time-dependent series of the solar illuminance 3 variables to estimate a power plant output at the end-time (3^{rd}) variable. Both networks were trained with previous 1 or 2 day 10-minute data series (i.e. samples), which provide the solar illuminance and corresponding power output values in the same location (Ostrava). Fig.3.-Fig.6. show the comparison of normalized power plant output day estimations, following the illuminance values. Both networks produce very similar results, despite their functionalities differs essentially. The power model outcome graph curves of Fig.5c. are nearly identical goings.

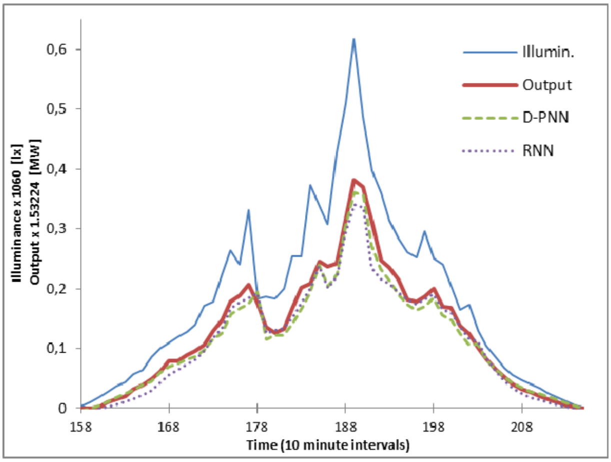


Fig. 3. RMSED-PNN = *0.0151*, $RMSE_{RNN}$ = *0.0184*

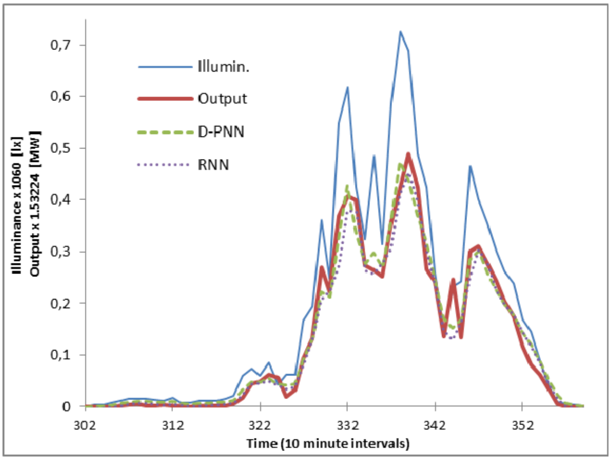


Fig. 4. RMSED-PNN = *0.0258*, $RMSE_{RNN}$ = *0.0275*

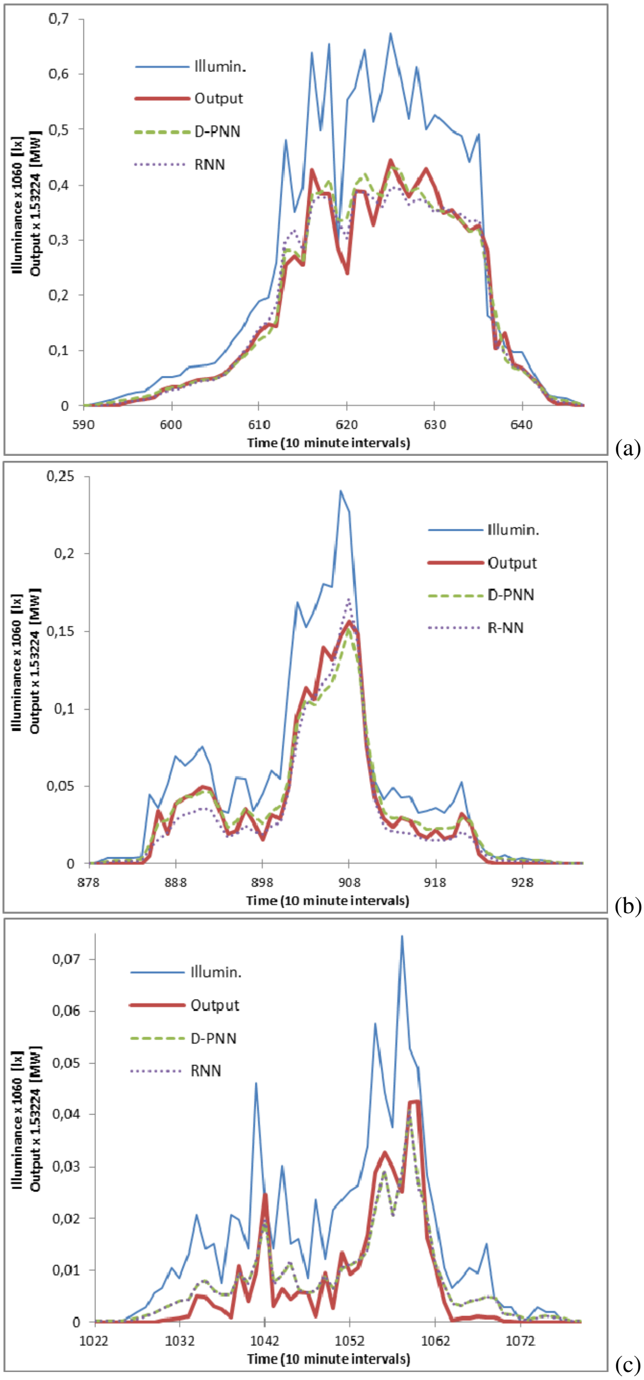


Fig. 5. a-c. $RMSE_{D\text{-}PNN}$ = 0.0242(a), 0.00694(b), 0.00369(c); $RMSE_{RNN}$ = 0.0234(a), 0.00747(b), 0.00374(c)

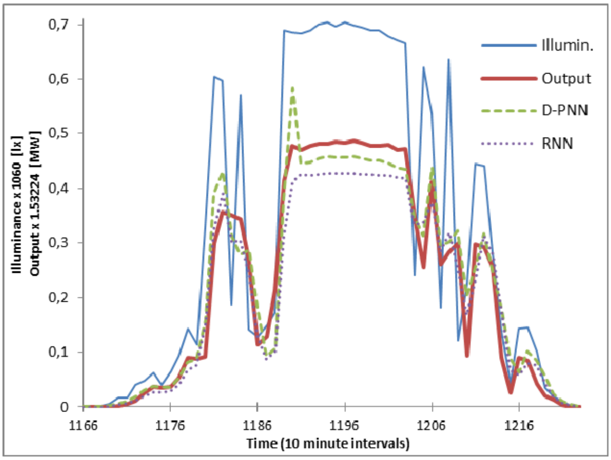


Fig. 6. RMSED-PNN = *0.0415*, $\mathrm{RMSE}_{\mathrm{RNN}}$ = *0.0440*

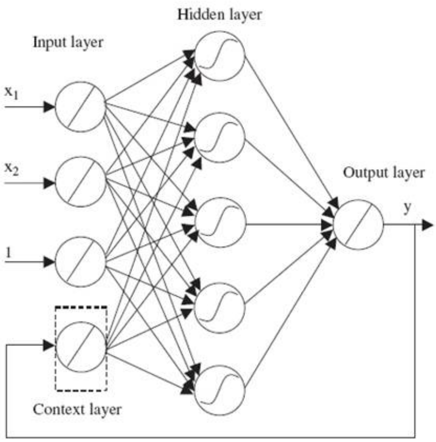
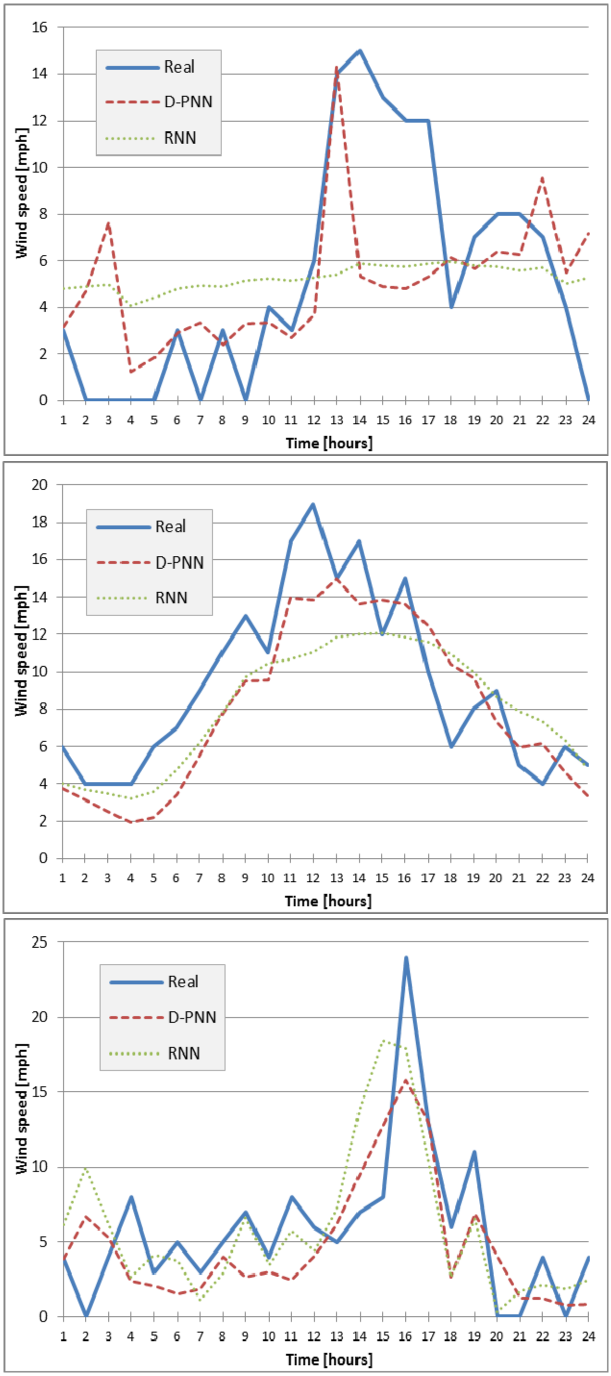


Fig. 7. Recurrent neural network

6 Wind Speed Model Experiments

The wind speed induces mainly a wind charger power output. There is usually available only its very rough forecast in a location. It could be modeled with reference to other weather variables forecasts, as meteorological predictions of this very complex dynamic system are sophisticated and not any time faithful, using simple neural network models. The fittest variables, which wind speed depends on, seem to be temperature, relative humidity and sea level pressure. The D-PNN and RNN models (Fig.8a-c.) apply 3 time-series of 3 state variables of 1 site locality, i.e. 9 input vector variables totally. Both networks were trained with previous 1 or 2 day hourly data series (24 or 48 hours, i.e. data samples), which are free online available [10].



(a)

(b)

(c)

Fig. 8. a-c. $RMSE_{D\text{-}PNN}$ = 4.287(a), 2.665(b), 3.576(c); $RMSE_{RNN}$ = 4.493(a), 3.188(b), 4.116(c)

7 Conclusion

The study compares 2 neural network models, which results are very similar, despite the fact their operating principles differs by far. This comparison indicates, both method outcomes are of a very good level, extracting from the provided input data a maximum of useful information. Both networks update the models daily, to respect a dynamic character of applied meteorological variables. D-PNN is a new neural network type, which function approximation is based on generalized data relations. Its relative data processing is contrary to the common soft-computing method approach, which applications are subjected to a fixed interval of absolute values. Its operating principle differs by far from other common neural network techniques.

Acknowledgement. This work has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and by the Ministry of Industry and Trade of the Czech Republic, under the grant no. FR-TI1/420 , and by SGS, VŠB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/58.

References

1. Atsalakis, G., Nezis, D., Zopounidis, C.: Neuro-Fuzzy Versus Traditional Models for Forecasting Wind Energy Production. In: Advances in Data Analysis, Statistics for Industry and Technology, pp. 275–287 (2010)
2. Cococcioni, M., D'Andrea, E., Lazzerini, B.: 24-hour-ahead forecasting of energy production in solar PV systems. In: 11th International Conference on Intelligent Systems Design and Applications (2011)
3. Hippert, H.S., Pedreira, C.E., Souza, R.C.: Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. IEEE Transactions on Power Systems 16(1) (2001)
4. Kuneš, J., Vavroch, O., Franta, V.: Essentials of modeling. SNTL Praha (1989) (in Czech)
5. Nikolaev, N.Y., Iba, H.: Adaptive Learning of Polynomial Networks. Springer (2006)
6. Nikolaev, N.Y., Iba, H.: Polynomial harmonic GMDH learning networks for time series modelling. Neural Networks 16, 1527–1540 (2003)
7. Prokop, L., Mišák, S., Novosad, T., Kromer, P., Platoš, J., Snášel, V.: Artificially Evolved Soft Computing Models for Photovoltaic Power Plant Output Estimation. In: IEEE International Conference on Systems, COEX Seoul, Korea (2012)
8. Prokop, L., Mišák, S., Novosad, T., Kromer, P., Platoš, J., Snášel, V.: Photovoltaic Power Plant Output Estimation by Neural Networks and Fuzzy Inference. In: Intelligent Data Engineering and Automated Learning - 13th International Conference, Natal, Brazil (2012)
9. Zjavka, L.: Recognition of Generalized Patterns by a Differential Polynomial Neural Network. Engineering, Technology & Applied Science Research 2(1) (2012)
10. National Climatic Data Center of National Oceanic and Atmospheric Administration (NOAA), http://cdo.ncdc.noaa.gov/qclcd_ascii/